

Appl. No. 09 / 287,478
Amdt. Dated: Apr. 16, 2003
Reply to Office action of Dec. 15, 2003

DRAFT

REMARKS-General

Two new claims have been added that correspond to functionality previously described under Detailed Description: Operation (referencing Fig. 4E) and have been concretely implemented in p 65-66 (simulatecore.pl) of the microfiche code listings.

With regard to item 2 of the detailed action:

The status of Jenkins (6,401,114 B1) as prior art is in doubt, since the provisional filing date of May 1, 1997 postdates first demonstration of the present invention (early April 1997), with the provisional filed almost one year to the day of said demonstration.

Jenkins (6,401,114 B1) would appear to claim dynamic linking of compiled object code within a server, as given by the implied definition of "logical components" within the following paragraph (Summary, P1, column 1, line 65 and ff), a technology that goes back to at least Bishop, et al (4,425,618):

This is accomplished by constructing logical components that are then called and bound into the program at run time, which alleviates the need to compile the program as a single unit. This action of "binding" logical program components together was not practical until the availability of programming languages like Java that allows this type of "late binding".

The preferred embodiment of the present invention does contain methods implemented within a Java program, however these "logic components" (.class files) are always executed in a fixed sequence. With regards to the use of a Unique Identifier (UI), in Jenkins its use is apparently restricted to retrieval of a "user profile" that is used to customize HTML template data, a different art from customizing circuit template data. The very nature of a customized "user profile" would tend to imply that the intent of Jenkins' UI is primarily to cache a login. There is no suggestion of using such a UI in managing Server resources, particularly in an environment of anonymous users.

If a user profile is employed, the user profile is identified by the user logging onto the server or by the server retrieving client identification by a technique such as cookies (information stored on the client). This process normally uses a template similar to that shown in FIG. 3, and replaces the "tag" with information provided in the user profile. The template shown in FIG. 3 is coded in HTML and displays the open/close HTML tags. The program using this template is requested by the client 12 and the server 22 for the information to be merged and published to the client 12.

PATH INFO/EXTRA and cookie mechanisms for maintenance of state in an HTTP/HTML session constitute prior art for the present invention, as described in "CGI Programming on the World Wide Web", which has been included by reference in the present application. However, WWW programming in general was a less-mature art in the April 1997 time frame and in particular CAD/CAM software that made primary use of a web browser interface (HTML or Java Applet with default privileges) was nearly non-existent. For example, "CAD" and "web browser" do not show up together in a simple search until Faybishenko (5,757,925, filed July, 1996). Consequently CAD/CAM and web application software development constituted nearly completely separate arts in

Appl. No. 09 / 287,478
Amdt. Dated: Apr. 16, 2003
Reply to Office action of Dec. 15, 2003

DRAFT

the 1996-7 timeframe, and the existence of someone skilled in both arts is a hypothetical. Indeed, to this day, the vast majority of CAD tools typically utilize connections with state, such as continuous TCP/IP. To a degree, Van Huben (5,950,201) unconsciously illustrates a widely held misconception at that time of the web as suitable only for delivering static documents and simple calculations, in that his disclosure relegates web access to just retrieval of "process and pseudo-process results". This is a deficiency he corrects in a subsequent application filed approximately one year later (but after the demonstration date of the present invention).

From the web technology viewpoint, it would seem that both Jenkins and the cited references for Montulli (6,134,592, filed Aug. 1997; but also 5,774,670, filed October 1995) envision cookies as a means to avoid the complexities of keeping client state on the server, or in generated web pages, not as a means to index complex server state.

With respect to "As per claim 3", this claim depends for its novelty on lower numbered claims.

With respect to "As per claim 4-6", as in previous replies, we have pointed out in previous responses that the "unique identifier" is associated with objects, not users, in Van Huben.

Our preferred embodiment employs a relational database to serve as the Control Repository. Each data object in the Data Management System (DMS) is assigned a unique identifier that permits all information about the object to be recorded and tracked by a multiplicity of relational tables. The physical data is stored using conventional storage management techniques which allow any type of data (text or binary) to be tracked in it's original form. The data may even reside on multiple platforms.

A recently discovered reference that seems more relevant is Bachman, et al (5,907,621), although Bachman embeds the token in the web page, rather than in a cookie.

Claim 4 depends for its novelty on previous claims, and with regards to claim 5, neither Jenkins nor Van Huben talk about the management of temporary files, specifically with regards to an Unique Identifier.

With respect to "As per claim 7", account limits exist in traditional operating systems (i.e. Unix, etc.), and perhaps in some client-server CAD systems in the prior art, but not in contemporary web-based CAD tools, to the best of the applicant's knowledge. Certainly the simulation count limits disclosed are not found in either Van Huben or Jenkins.

With respect to "As per claim 8", van Huben does not teach lowering of process priority based on usage counts. Van Huben's resource/attribute tables would appear directed at preventing deadlock by enabling or disabling entire processes according to resource availability, rather than adjusting process priority.

Appl. No. 09 / 287,478
Amdt. Dated: Apr. 16, 2003
Reply to Office action of Dec. 15, 2003

DRAFT

With respect to "As per claim 9", van Huben does not mention circuit synthesis (this was part of Burrows ('117)). In any event the novelty of claim 9 rests on the continuity of design data kept on a server indexed by the Unique Identifier.

With respect to "As per claim 10", column van Huben (201) 18:20-25 refers to BOM email status(?), and one of the salient benefits of the present invention is the elimination of the need for a "Data Manager", since the present invention newly solves the different problem of anonymous, unmanaged use of a complex simulation tool. This paragraph in van Huben actually contrasts quite strongly with the intended use of the present invention. 6:54-27 again emphasizes data management over the internet and the execution of "batch jobs", but not over an anonymous, interactive, HTTP/stateless communication medium.

With respect to "As per claim 11", the privileges described herein are not necessarily assigned by an account manager. They be derived from a "ticket", or by means of the link the user used to reach the first interface page, etc.

With respect to "As per claim 12", while van Huben does disclose methods for operating across a TCP/IP network such as the internet, he does not disclose methods for management of state using a stateless (i.e. HTTP) connection medium. His discussion of WWW/Internet access focuses on retrieval of job status of previously initiated processes, and although "authorized users" can manually initiate DILPs (library processes), the results must apparently be retrieved as separate step and does not constitute a system for anonymous, automatically managed interactive simulation.

With respect to "As per claims 13" and "14", Jenkins mentions chemical analysis but nowhere mentions anything electrical or circuit or simulation. Cookies are but one means to the storage of a Unique Identifier (Jenkins does not discuss PATH_INFO or PATH_EXTRA for example), and in any event the importance lies in what is stored and how the stored data is updated, not how it is stored.

Appl. No. 09 / 287,478
Amdt. Dated: Apr. 16, 2003
Reply to Office action of Dec. 15, 2003

DRAFT

Conclusion:

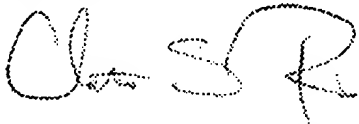
We still hold that the present application Solves a Different Problem from any of the references attached in the office action and that the field is a Crowded Art in which the methods disclosed are a significant advance over what came before. In particular the dynamically and automatically assigned, transient Unique Identifier is a novel art with respect to the field of Computer Simulation.

I will retain profession counsel to discuss redrafting the claims to further clarify the distinctions from the prior art.

Conditional Request for Constructive Assistance:

The applicant has amended the Background and the Claims so that they are proper, definite and define novel methods which are also unobvious. If, for any reason this application is not believed to be in full condition for allowance, applicant respectfully requests the constructive assistance and suggestion of the Examiner pursuant to M.P.E.P. S2173.02 and S707.07(j) in order that the undersigned can place this application in allowable condition as soon as possible and without the need for further proceedings.

Very respectfully,



Christian S. Rode, Applicant Pro Se
2412 Stearns Hill Rd.
Waltham, MA 02451
Tel. 781-899-4322 Fax (same, but call first)